DOCUMENT RESUME

ED 427 792 IR 057 299

AUTHOR	Boatright, George S.
TITLE	LATEX for Librarians.
PUB DATE	1999-03-02
NOTE	16p.
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Oriented Programs; Computer Software Development; Electronic Text; Information Processing; Information Technology; Mathematics; Technical Mathematics; Technical Writing; Troubleshooting; *Word Processing
IDENTIFIERS	*LATEX Computer Program

ABSTRACT

This paper presents the basics of "LATEX," a computerized typesetting program popular with mathematicians and other writers of technical papers, and compares it to conventional word processing programs. LATEX, which requires a text editor such as WordPad or NotePad--the text editor is used to produce and input file which the LATEX program processes--can be used for articles, letters, reports, and books. This paper demonstrates how to use LATEX, including commands for simple formatting, for producing references and a bibliography, and for customizing LATEX by defining new commands. Sections on troubleshooting and recommendations are also provided. LATEX can be downloaded or purchased from other commercial sources. (AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 427 792

LaTeX for Librarians

George S. Boatright

March 2, 1999

Abstract

The author presents the basics of LaTeX, a computerized typesetting program, and compares it to conventional word processing programs. A short bibliography is included.

1 What is LaTeX?

LaTeX is a computerized typesetting program popular with mathematicians and other writers of technical papers. It can be used for articles, letters, reports, and books. Unlike popular word processing programs, it is not a WYSIWYG (what you see is what you get) program. In LaTeX whether you put one space or ten spaces between words the end result is the same. It can be compared to computer programming, although a background in programming is not necessary.

2 Why bother with LaTeX?

It is not the purpose of this article to debate the relative merits of document preparations systems. Any writer who makes extensive use of formulae and tables will come to appreciate the power of LaTeX. Although the conventional programs can handle formulae, tables, etc., they have certain limitations.

BEST COPY AVAILABLE

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
G.S. Boatright
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ERO 57299

3 A little historical background

Donald Knuth [5] a computer scientist, wrote a program called TEX because of his frustration with conventional typesetting of mathematical formulae. Leslie Lamport [7] produced LATEX, a version of TEX, that many writers find easier to use. The latest version is LATEX2ε and all references in this article are to that version.

4 How can I get LATEX?

You can download LATEX from the Internet or buy it from different sources. Details will be provided in another section. You will also need a text editor such as WordPad or NotePad. Any editor that produces ASCII text will be sufficient

5 Getting started

As mentioned above, LATEX is not like conventional word processing programs where you what you see on your monitor is very similar to the printed output. You must use your text editor to produce an input file which the LATEX program processes. If there are no errors, the program produces a dvi ("device independent") file. A previewer program will allow you to see how the printed output will appear.

Every input file needs a minimum of three commands:

- `\documentclass{ }`
- `\begindocument`
- `\enddocument`

The `\documentclass` can be an `article`, `report`, `book`, `letter` or `slides`.

There are ten special characters that you must use with caution:

BEST COPY AVAILABLE

$ % & ~ _ ^ \ { }

Seven of these symbols are easy to produce by putting a \ in front of them. ~, ^, and \ will be discussed later.

You type sentences just the way you would at a typewriter or your regular word processor but with an important difference. One space or several spaces between words and sentences will not produce extra spaces in your output. \, between characters will produce a small amount of extra space if necessary.

Compare space and space as an example. The second "space" is typed `space\,\,` in the input file.

A period causes LaTeX to produce some extra space. Compare these two sentences.

She has a PhD. in education.
She has a PhD. in education.

The second sentence has a \ after the abbreviation to correct the spacing. The command `\ldots` is used for an ellipsis. Three periods will cause the wrong spacing. Compare these sentences.

I never thought about . . . before.
I never thought about ...before.

The second line has `\ldots` after about.

Another way to control spacing is with `\@`. Compare these sentences.

I learned to program in FORTRAN. I hated punched cards.
I learned to program in FORTRAN. I hated punched cards.

The second senctence has `\@` between the N and the period. This is used whenever the last word in a sentence ends with a capital letter.

A blank line makes the program begin a new paragraph.

It's easy to change the appearance of text. `\textbf{ }` causes the text within the { } to appear in **bold face**. Likewise `\textit{ }` causes the text to change to *italics*. Or if you prefer sans serif that's easy with `\textsf{ }`. You can simulate the text produced by a text editor with

the `\verb` command. For example, to produce \ you would type `\verb+\+` in your input file. Underlining is simple with the `\underline{ }` command.

It's also easy to change the size of type. In the `\documentclass[ pt]{article}` you can change the default of 10pt to 11 or 12pt. Within text you can alter the size with different declarations:

`\tiny` followed by text produces very small type

`\large` followed by text produces much larger type.

There are four different dashes. A single dash results in - while double and triple dashes result in – and —, respectively. If you want the minus sign, you must type `$-$` in order to get −.

LaTeX can produce a variety of accents and special symbols. For example:

María	señor	bilingüe
Mečiar	leçon	élevè
ı	ȷ	£
Œ	Ø	ß

If you look closely, you will see that the dots have been removed from the i and the j.

The key above the tab key on most keyboards will produce a single left quote `. Using it twice produces the left quote ``. The key to the left of the enter key produces a single right quote '. Using it twice gives a double right quote ''. Using the shift key and this key will also produce the ".

It's possible, and usually preferable, to offset, or *display*, long quotations. One way to do this is with `\begin{quote}`. For example

> How do you pronounce LaTeX?
> The usual way to say it
> is lay as in hey
> and tech as in high tech[1].

A double backslash causes the program to start a new line. It is omitted for the last line. `\end{quote}` is necessary to end the quotation.

[1] The author apologizes for his bad verse.

BEST COPY AVAILABLE

Of course, quotations frequently require footnotes. `\footnote{ }` will produce a numbered footnote at the bottom of the page. The text of the footnote goes between the curly braces. LaTeX has the ability to make a bibliography and an index. These features will be discussed in another section.

Itemized lists with "bullets" are popular features on word processing programs and LaTeX can also produces lists with the `\begin{itemize}` command.

- I think bullets are a waste of toner.
- I think they should be called bullet holes.

You can also produce numbered lists with the `\begin{enumerate}` command. We can easily change one of the lists above to a numbered list with this command. Every input file needs a minimum of three commands:

1. `\documentclass{article}`
2. `\begindocument`
3. `\enddocument`

The `\begin{enumerate}` command also makes it possible to produce lists within lists, or *nested* lists. The syntax to produce lists is

```
\begin{itemize}
\item
\item
\end{itemize}
```

6 A little more advanced

LaTeX can produce tables or forms in the tabular environment. The example above showing a variety of symbols and accent marks was produced with this environment. The text in the input file looks like this:

```
\begin{tabular}{lrc}
Mar\'{\i}a & se\~{n}or &biling\"{u}e \\
Me\v{c}iar & le\c{c}on & \'{e}lev\`{e}\\
\i & \j& \pounds\\
\OE & \O & \ss
\end{tabular}
```

The {lrc} indicates that there are three columns in the table and that they are left justified, right justified, and centered. Of course, you are free to choose how you want the columns justified. The & separates each column and the \\ causes the program to skip to the next row. It is omitted for the last row. It's easy to add horizontal and vertical lines to tables. \hline will add a horizontal line wherever necessary. Changing {lrc} to {|l|r|c|} will produce vertical lines between the columns.

María	señor	bilingüe
Mečiar	leçon	élevè
ı	ȷ	£
Œ	Ø	ß

An *array* is similar to a table except it is used in the math mode. The commands are surrounded by \(and \). Arrays are used for mathematical symbols and numbers. The simple example above would not be appropriate for an array. The syntax is very similar to that of tables.

$$\begin{array}{lrc}
\sin x & x & \pi \\
12 & \ln x & 3 \\
\tanh x & 0 & 1
\end{array}$$

This text produced the example above:

```
\( \begin{array}{lrc}
 \sin x & x & \pi \\
12 & \ln x & 3 \\
\tanh x & 0 & 1
\end{array}\)
```

Arrays are frequently surrounded by *delimiters*. Adding \left(before \begin{array} and \right) after \end{array} will produce:

$$\begin{pmatrix} \sin x & x & \bar{\pi} \\ 12 & \ln x & 3 \\ \tanh x & 0 & 1 \end{pmatrix}$$

A variety of delimiters can be used. You must use `\left` and `\right` but they do not have to match. You can make a delimiter disappear with a period after the `\left` or `\right`.

The real advantages of LaTeX become obvious when dealing with arrays, or matrices, and equations. Conventional word processing programs treat them as "objects" which are inserted into the text. These objects are more difficult to manipulate and also take up more disk space. Mathematical equations can be used within text or displayed as a quotation. Within text the equation is surrounded by dollar signs. For example, $y = \log x$ is typed `$y = \log x$`. You must surround displayed equations with `\[` and `\]`. Here is Euler's famous equation:

$$e^{\pi i} + 1 = 0$$

You can number equations and then refer to them later. To produce a number for the above equation, we would have to type

```
\begin{equation}\label{euler}
e^{\pi i} + 1 = 0
\end{equation}
```

You could now refer to the equation by its key `{euler}`. The key can be letters or numbers. See the following section **Some bibliographic details**.

Superscripts and subscripts are easy to produce with ^ and _. It's not necessary to point and click to produce x^2 or y_i. You may have to use braces for more complicated expressions. To produce x^{n+1} within your text you would type `$x^{n + 1}$`. To add a subscript to this expression you would just change it to `$x^{n + 1}_{n + 2}$` to get x^{n+1}_{n+2}.

Fractions can go in text by using /. $(a + b)/ab$ within text is typed `$(a + b)/ab$`. More complicated fractions are usually displayed:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

The `\frac` command is used for an equation like the one above. It has two arguments, the numerator and the denominator.

BEST COPY AVAILABLE

Square and other roots are produced with the `\sqrt` which has an optional argument for other roots. $\sqrt{2}$ is produced by typing `\sqrt{2}`. For $\sqrt[3]{2}$ it would be necessary to type `\sqrt[3]{2}`.

7 Some simple formatting

If you are required to use double or 1.5 spacing, you will have to make some changes in the *preamble* of your input file. This is the section between `\documentclass` and `\begindocument`. Placing this command in the preamble will produce doublespacing between lines:

```
\renewcommand{\baselinestretch}{2}
```

Changing the 2 to 1.5 will produce a space and a half between lines. You can control indentation and spacing between paragraphs with these commands:

```
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
\setlength{\parindent}{0pt}
```

The upper command produces extra space between paragraphs. The lower one suppresses indentation at the beginning of paragraphs. Normally, there is no indentation of the first paragraph at the beginning of a section.

Different margins also require added statements in the preamble. If you wanted a 1.5 inch margin on the left, you would add these lines in the preamble:

```
\setlength{\oddsidemargin}{.5in}
\setlength{\evensidemargin}{.5in}
```

.5 is used because LaTeX measures distances 1 inch from the left edge. In a similar fashion it measures 1 inch from the top edge. The `\setlength` command is also used to change the top margin if necessary.

Centering text is simple. The process is similar to that of displaying quotations. It begins with `\begin{center}` followed by text and ends with `\end{center}`. The double backslash after a word will cause the program to begin a new line.

If you want to produce the LaTeX logo,
You must type it like this:
`\LaTeX\`
It's case sensitive!

If you want to start a new paragraph after the centered text, you must skip a line after `\end{center}`.

In a similar fashion, it's possible to produce text that is aligned with the left or right margin with `{flushright}` (or left).

If you want to produce the LaTeX logo,
You must type it like this:
`\LaTeX\`
It's case sensitive!

It is easy to divide your text into columns by adding `[twocolumn]` as an option to the `\documentclass`. As mentioned above, this is where you can change type size. You can have several options to change the default format. In the example below, we have changed the type size to 11 and divided the text into columns.

```
\documentclass [twocolumn,11pt]{article}
```

LaTeX automatically produces a centered arabic number at the bottom of the page. You can change this with `\pagenumbering{ }` before the text you want numbered. `Roman` in the braces will produce an upper case roman numeral for the page number. Other options are `roman` for a lower case roman numeral, `alph` for lowercase letters, and `Alph` for uppercase letters. If you need to begin numbering with some number other than 1, you can do so with the command

```
\setcounter{page}{ }
```

where the number you wish to start with goes within the braces. You can also change the location of the page numbers. Adding these lines to the preamble

BEST COPY AVAILABLE

```
\pagestyle{myheadings}
\markright{ }
```

will produce 1 at the bottom of the first page and the rest of the numbers will be flush right in the upper corner. In this example you could also add text within the braces after the `\markright` command. This example is for one sided printing where the pages are considered to be right–hand. For two sided printing you would have to add `twoside` as a document class option and use `\markboth{ }`. If you do not want any page numbers, change the `\pagestyle` option to `empty`.

If you wish to divide your article into numbered sections and subsections, you can use the `\section{ }` and `\subsection{ }` commands. The title of the section or subsection goes within the curly braces. The program will produce the numbers.

You can make a title page with a few simple commands. The title of this article was created with this text:

```
\title{\LaTeX\ for Librarians}
\author{George S. Boatright}
\maketitle
```

The title and author go within the braces. You can create an abstract with

```
\begin{abstract}
```

text of abstract

```
\end{abstract}
```

8 Some bibliographic details

LaTeX can easily produce numbered references and a bibliography. The simplest way is with these commands:

```
\bibliographystyle{plain}
\begin{thebibliography}{99}
\bibitem
```

```
\bibitem
\end{thebibliography}
```

The syntax for the actual citation is `\bibitem{ }` followed by the bibliographic information in whatever style you choose. The text within the curly braces, or *key*, is usually an author's last name, but it can be any letters or numbers. If you wanted to refer to an author's work within your text, you would write the name as you had listed it within the braces followed by `~\cite{ }`. For example, at the beginning of the article there is a reference to Lamport. The actual text is `Lamport~\cite{Lamport}` which produces a numbered reference. The `plain` bibliography style follows the format in van Leunen [9]. Other choices are `unsrt`, (entries are in order of citation). Goossens [1] gives more details. The `{99}` should be a number or bit of text at least as wide as the widest item label in the list of references. This would suffice for a bibliography of between 10 and 99 items.

One option is the use of a *label* after the `bibitem`. This is useful if one author has more than one item in the bibliography. It is enclosed in `[ ]`. You can put any number or letter you wish within the brackets. Also the key should be slightly different to allow for multiple citations. For example, you could have `\bibitem{Knuth}` and `\bibitem[5a]{Knuth2}` to cite more than one work by Knuth. In the text, if you wished to refer to both works, you would have `Knuth~\cite{Knuth,Knuth2}`.

LaTeX also has the capability of producing a bibliographic database with the `\bibliography{ }` command. The name of of one or more bibliographic files with a `bib` extension goes within the braces. For writers who cite the same works in different articles a bibliographic database will save time in the long run. See Lamport [7] or Kopka [6] for details. The name of the program is BIBTeX.

There are two ways to generate an index or a glossary for your documents. You can use the MakeIndex program which is usually included in the different packages along with documentation. The other way is with the `makeindex` environment. This environment will produce a two column index. It follows this general form:

```
\begin{theindex}
```


BEST COPY AVAILABLE

```
\item
\item
\end{theindex}
```

The `\item` precedes each entry in the index. You can also have `\subitem` and `\subsubitem`.

9 Customizing LaTeX

You can customize LaTeX by defining new commands. This can save typing if there are certain commands you frequently use. If your document contains many lists, you could define new commands for them. Instead of typing `\begin{itemize}` and `\end{itemize}`, you could define two new commands:

```
\newcommand{\bi}{\begin{itemize}}
\newcommand{\ei}{\end{itemize}}
```

It's usually best to put new commands in the preamble along with a comment or reminder about what they do.

In a similar fashion you can define new environments. We have already seen examples of environments related to lists and tables. The language follows this structure:

```
\newevironment{   }{    }{    }
```

The name of the new environment goes within the braces immediately following `\newenvironment`. This is followed by the beginning definition and the ending definition.

10 Internet resources

The TeX Users Group's website (http://www.tug.org) is a good starting place. You can get information on downloading LaTeX or obtaining it from commercial sources. There are versions for Unix, DOS, OS2, Macintosh, and Windows95/NT. This article was produced with MikTeX.

BEST COPY AVAILABLE

If you join the users group, you will receive a set of CD-ROMs with the LaTeX program and extensive documentation. It is even possible to run the program from one of the CD-ROMs. As of this writing, a one year membership is $65.00 and includes a subscription to the group's journal. The American Mathematical Society's website (http://www.ams.org) also has some information.

11 Troubleshooting

The beginning LaTeX user will probably get an error message after running the program on an input file. For example, if you type `$` without a backslash in front of it, you will get a message of this sort:

```
!Missing $ inserted
 <inserted text>
                $
l.369
?
```

At this point, you can enter `h` to get a help message, `r` to continue running the program, or `x` to stop the program. You can also enter `e` which will bring up your text editor or `?` which will give you a list of possible actions.

Some common mistakes are:

- A missing `\end{ }` command
- A missing or mismatched brace or bracket
- A missing `\` before the special symbols
- Misspelled commands

It's usually best to run your input file frequently to avoid too many error messages at any one time.

Besides error messages LaTeX produces warnings. The program will still run and produce a dvi file. A common warning is `Overfull \hbox`. The program could not find a good place to break a line. You can wait until your final output to take care of these problems with `\linebreak` or `\pagebreak` commands. `\newpage` will cause the program to begin a new page.

12 Some recommendations

If you decide to try LaTeX, a good starting point is with two files usually found in a distribution set. Look for `small2e.tex` and `sample2e.tex`. These are small files that you can run to produce `dvi` files. You can also use them as a basis for your own files. Remember that the program will ignore any text with `%` before it. Also look for a *local user's guide* in your distribution set. At least some of the documentation will be in the form of `dvi` files. You can click on them and the previewer will allow you to read them.

The `small2e.tex` and `sample2e.tex` files have information about LaTeX. A good exercise is to try to duplicate the text of these files yourself. *The Not So Short Introduction to LaTeX 2ε*, which has much more information, can be downloaded as a dvi file from the TeX Users Group website.

An easy way to submit a file in the Windows environment is to click on the start icon and then click on the run icon. This is one possiblity that you might have in the run "box."

```
c:\texmf\miktex\bin\latex a:\small2e.tex
```

In this example, the input file is on the A drive. Click OK and another window will appear with any error messages. If all goes well, you will have a `small2e.dvi` file in the bin directory. If you click on this file, the previewer will display the output on your monitor. You can also print from the previewer. You can also use the minimize feature to go back and forth between the text editor and the previewer.

BEST COPY AVAILABLE

References

[1] Michel Goosens, Frank Mittelbach, and Alexander Samarin, *The LaTeX Companion* Addison-Wesley Publishing Company, Reading, MA, 1994

[2] George A. Gratzer, *Math into LaTeX an Introduction to LaTeX and AMS LaTeX* Birkhauser, Boston, 1996

[3] David Griffiths and Desmond J. Higham, *Learning LaTeX*, SIAM, Philadelphia, 1996.

[4] Alan Hoenig, *TeX Unbound: LaTeX & TeX Strategies for Fonts, Graphics, more* Oxford University Press, New York, 1998

[5] Donald E. Knuth, *Computers and Typesetting* Addison-Wesley Publishing Company, Reading, MA 1986

[5a] Donald E. Knuth, *The TeXbook* Addison-Wesley Publishing Company, Reading, MA, 1986

[6] Helmut Kopka and Patrick W. Daly, *A Guide to LaTeX 2ε Document Preparation for Beginners and Advanced Users* 2nd. ed., Addison-Wesley Publishing Company, Reading, MA 1995

[7] Leslie Lamport, *LaTeX a Document Preparation System User's Guide and Reference Manual* 2nd. ed., Addison-Wesley Publishing Company, Reading, MA 1994

[8] Tobias Oetiker, *The Not So Short Introduction to LaTeX 2ε* available as a dvi file from http://www.tug.org ↓999

[9] Mary-Claire van Leunen, *A Handbook for Scholars* revised edition, Oxford University Press, New York, 1992

BEST COPY AVAILABLE